A synthesis of Quality Criteria for requirements Elicitation Methods

A SYNTHESIS OF QUALITY CRITERIA FOR REQUIREMENTS ELICITATION METHODS

Peter Bollen
Maastricht University, Faculty of Economics and Business Administration, Maastricht, the Netherlands

John Simons
Groningen University, Faculty of Management and Organization, Groningen, the Netherlands

Abstract. In this article we define quality criteria for requirements elicitation methods. These criteria are a synthesis of quality concepts and ideas in the fields of software engineering, information system development methodologies, conceptual modeling and requirements engineering. They underline the development of requirement elicitation methods from traditional methods (EER,ORM) to modern methods (UML,NLM).

Keywords: Requirements engineering, requirements determination, requirements elicitation, criteria for RE approaches.

1. Introduction

The development of effective information systems requires a thorough requirements elicitation process [8]. Improvements in the requirements elicitation process will lead to improved systems development efforts [6].[1]

Many publications in the field of information systems acknowledge the importance of the Requirements Elicitation stage in information systems development projects, but only a few contain explicit criteria [18, 40] to which a requirements elicitation approach must comply. The papers in the literature that address these (sets of) criteria are scattered around the fields of *software engineering* [23, 25], *(information) systems development methodologies* [46], *conceptual modeling* [33] and *requirements elicitation* [42, 52].

Most requirements elicitation methods do not assess the compliance to objective criteria in the literature, partly due to the absence of criteria that can operationally be applied by RE analysts.

[1] Requirements Determination (RD), Requirements Elicitation (RE) or Requirements Analysis (RA) is considered to be one of the most critical activities in an information systems development project [21]. Requirements determination, -elicitation or -analysis contributes to a large extent as a source of information systems failures [8, 20, 49], respectively systems that do not satisfy users. In the remainder of this article we will refer to RD, RE, or RA as *Requirements Elicitation*

Hickey and Davis [21] discuss a Unified Model of Requirements in which they distinguish types of requirements that determine the selection of the technique for the next stage in a requirements elicitation process. They partition the available techniques into a class of applicable techniques and a class of non-applicable techniques. In a real-life environment, however, the suitability of a specific requirements elicitation technique, for a specific phase in the RE process, must be captured on a continuous scale, rather than be positioned as either *applicable* or *not applicable.*

Pitts and Browne [40] discuss a more specific issue in RE, namely, stopping criteria for the analyst. They performed an empirical analysis in which the variables *analyst experience* and the *type of cognitive stopping rule that will be applied* are linked to the exogenous variable: *requirements elicitation outcome*. Pitts and Browne conclude that the (overall) quantity and the depth of the requirements is statistically related to the type of stopping rule that was applied by the subjects in their study. The stopping rules that were considered in their study (*magnitude threshold, difference threshold, mental list, representational stability*) have in common that they require the analyst to decide when to stop. In the research design no formalized stopping rule that is embedded in a RE approach was considered.

Grünbacher et al. [18] propose to design repeatable quality assurance steps and use them with proved collaborative processes. Grünbacher et al. distinguish *pre-*, *in-* and *post-process* quality assurance steps. These authors conclude from a feasibility study that the application of these quality assurance techniques reveals defects and therefore has a positive effect on the quality of the resulting requirements specifications. However, they do not provide a specific (set of) quality assurance (QA) technique(s).

This paper is organized as follows. In the next paragraph we discuss and summarize the literature on (partial) quality criteria for RE methods. In the following paragraph we synthesize these criteria into four coherent and consistent criteria, which are completeness, efficiency, formality and domain richness. We argue that they apply to three different aspect of RE, being the way of modeling (the product of RE), the way of working (the operational process of RE) and the way of controlling (the managerial process of RE). Finally we show that the development of requirement elicitation methods (from traditional EER to modern NLM) is in line with this synthesis of quality criteria.

2. Partial quality criteria from the literature

We present a literature review on quality criteria for specifications, (software and information systems) requirements and conceptual schema's. We analyze these criteria for their attribution to the *way of modeling* (the specifications as a product) or the *way of working* (the process that creates the specification) [41] or the *way of controlling* (the project and quality management involved in creating a specification) [48].

2.1 Criteria for Software Engineering Methods

Khwaja and Urban [25] present one set of criteria for the specification (way of modeling) and another set of criteria for the specification technique (way of working) from a literature research in the field of software engineering. They map the specification technique criteria to specification criteria as given in table 1. Most criteria that are listed under 'specification criteria' are defined in the IEEE Std. 830 sets of criteria [23] that we will discuss later in this paper,

Table 1. Criteria for software specification techniques and mapping onto specification criteria (adapted from table 3 in Khwaja and Urban [25:p.592-593] and specification criteria according to Boehm [3].

Technique criterion (Khwaja and Urban , 2002)	**Specification criteria (Khwaja and Urban , 2002)**	**Specification criteria (Boehm, 1984)**
Expressive adequacy	Understandable, appropriate, minimal	Completeness
Constructability	-	Consistency
Scope of specifications	complete	Feasibility
Level of formality	Unambiguous, consistent, complete, verifiable, validateable	Testability
Formal foundation	Unambiguous, consistent, complete, verifiable, validateable	
Extent of applicability	-	
Ease of use	-	
Help support	-	
Integrated environment & Tool support	-	
Specification organizational support	Understandable, modifiable	
Support for maintainability	Modifiable, traceable	
Executable	Understandable, unambiguous, consistent, complete, correct, verifiable, validateable	
Tolerance for incompleteness	Verifiable, validateable	
Multiple views	Understandable	
Notational simplicity & flexibility	Understandable	
Internal verification support	Unambiguous, complete, consistent, verifiable	
External validation support	Correct, validateable	
Support for other development phases	Traceable	
Support for document generation	Understandable	

The first technique criterion that they mention is *expressive adequacy* that refers to the expressive capability of a technique to improve the understanding, appropriateness and minimality of a requirements specification. Their second specification technique criterion is *constructability* that refers to the availability of a technique for RE. Their third criterion *level of formality (or formal foundation)*

that is defined as*: 'High level of formality in a specification technique may help defining, precise, unambiguous, consistent, complete and verifiable specifications.'* [25]. Their next criterion is *extent of applicability* which refers to the range of domains in which the technique can be applied. The next relevant criterion is *specification organizational support*: 'Good specification organization helps in controlling complexity and enhancing understandability. This criterion may also help in modifying specifications.' Their next criterion is *support for maintainability* which refers to modifiability and traceability tools. The next specification technique criterion is called *executable* and measures the extent in which a specification technique can create an operational model of the specifications that leads to a specification that complies with all listed criteria for a specification. *Tolerance for incompleteness* is a criterion that measures how *'...incomplete specifications may help testability at various stages of specification development.'* [25:p.593]. The criterion *multiple views* measures the extent in which a specification technique can create multiple views of a specification. The criteria *internal verification support* and *external validation support* measure the extent in which a specification technique has provisions for internal verification of the resulting specifications, respectively has provisions for validating the specification against test cases to ensure correctness. Other listed criteria (ease of use, help support) overlap with these criteria.

Boehm [3: p.206] defines *completeness* in the context of requirements and design specifications as the extent in which all of the specification part is present and fully developed. Boehm [3] defines *consistency* as internal consistency: *' items within the specification do not conflict with each other*' [3: p.77-78]. Boehm defines *traceability* as a sub-criterion within consistency: *'items in the specification have clear antecedents in earlier specifications or statement of system objectives.'* [3: p.78]. *Feasibility* is concerned with the economics of an IS development and implementation project and *testability* refers to the properties of a software implementation. The latter two criteria are outside the scope of quality criteria for a requirements specification in this article.

Both Khwaja & Urban and Boehm argue the plausibility of their individual criteria, however do not discuss the coherence and the consistency of their criteria.

2.2 Criteria for Information Systems Development Methods

In the field of information systems development methods the systems life cycle [30:p.393-398]: *Systems analysis*, *systems design*, *programming, testing, conversion* and *production and maintenance* plays a major role. The requirements elicitation is the second activity after the feasibility study in the stage *systems analysis*. In table 2 the criteria for information systems development methods are listed (with focus on the way of working and the way of controlling) as given by Wysocki and Young [51].

Table 2. Criteria for Information systems development (methods)

Author	**Wysocki and Young (1990)**	**Essink and Romkema (1989)**
criteria	Efficiency	User participation
	Communications	Maintainable
	control	Specification must be an expression of the real domain requirements
	Documentation	IS must be built according to specifications
	Role definition	Efficient development process
	Consistency	

Wysocki and Young [51] give the following benefits or 'goals' of system development methodologies (SDMs). The first criterion (goal) of Wysocki and Young [51: p.298] referring to the way of working is *efficiency*, which they describe as *'a certain degree of guidance and direction can definitely improve efficiency '*. Their second criterion referring to the way of controlling is *communications. 'As SDM usually designates specific points during a project at which certain facts are to be communicated to others. This communication could involve both specific members of the project team and outside parties'* [51: p.299]. Their third criterion is *control*. This criterion is closely linked to the criterion of communication since it requires SDMs to have a monitoring system for go/no-go decisions at specific milestones. Another 'goal' of a SDM according to Wysocki and Young is *documentation*: *'..documentation can include users'manuals, systems overview information, operator instructions, maintenance programmer background information, restart and error recovery instructions, and more.'* [51: p.300]. The 'goal' *role definition* '..serves to define the roles of everyone in the system.' [51: p.300]. The final goal that is mentioned by Wysocki and Young is *consistency,* it emphasizes the availability of a readily repeated process to build information systems.

Essink and Romkema [13] give a number of demands for information systems development. The first demand is that the *user participation* in the development process must be optimal. The second demand is that the resulting information system must be easy *maintainable*. The third demand is that the information system's specification must be an *expression of the real domain requirements.* The fourth demand that is given by Essink and Romkema is that an information system must be built according to the latter specifications. Finally, Essink and Romkema demand that an information systems development process has to be performed in an *efficient* way.

These methods address (implicit) quality criteria for the way of working mainly. The debate on quality criteria in the MIS-field was very intensive during the 80-ies and has lost attention due to the absence of a significant link between 'good' methods and 'good' system specifications '*Methods are nothing,*

methodizing counts'. Also in this field coherence and consistency of a set of criteria is unadressed.

2.3 Criteria for Conceptual Modeling languages

The conceptual schema research field emerged in the 1970's and provided a counter-force against the predominance of implementation models, e.g. record-based models [38: p.327], respectively specific software development approaches, e.g. object-orientation [12:p.184]. The conceptual schema approach is based upon a transformation mechanism [32] in which instances of the external schema are transformed into a conceptual schema (that is expressed in a given conceptual modeling language), and subsequently the conceptual schema can be transformed onto an internal or 'implementation' schema [32:p.69].

The criteria in table 3 refer to the properties that a conceptual modeling language must possess. They focus on the way of modeling. In table 3 we summarize the criteria given by Nijssen [38], Loucopoulos [34] and Halpin [19].

Table 3. Criteria for Conceptual Modeling (languages)

Author	**Nijssen (1977)**	**Loucopoulos (1992)**	**Halpin (2001)**
criteria	Formality	Implementation independence	Expressibility
	Completeness (in terms of describing the UoD)	Abstraction	Clarity
	Easy to formulate	Formality	Simplicity and orthogonality
	Easy to understand	Constructability	Semantic stability
	Easy to change	Ease of analysis	Semantic relevance
		Traceability	Validation mechanisms
		Executability	Abstraction mechanisms
			Formal foundation

Nijssen [38:p.327] gives a number of criteria for a (conceptual modeling) language in a way that *' …a language in which it is possible to describe formally, completely, easy to formulate, easy to understand and easy to change the Universe of Discourse covered by a set of information systems.'*

Loucopoulos [33] and Halpin [19] list the criteria *constructability*, respectively *abstraction mechanisms* that both stand for capability of a conceptual modeling language to handle scopes or views on a large global schema. Loucopoulos and Halpin both give a formality criterion (*formality* respectively *formal foundation*) that measures the *(un)ambiguity* of a conceptual schema. Loucopoulos defines *ease of analysis* as the extent in which a conceptual schema can be analyzed for *ambiguity, completeness* and *consistency.* This criterion partially overlaps with Halpin's *clarity*. Loucopoulos, furthermore gives the criterion of *implementation independence* which is similar to Halpin's criterion of

semantic relevance. Both criteria refer to the *conceptualization* principle [24] and state that only conceptually relevant details need to be modeled. Loucopoulos gives the criterion of *traceability* which refers to the ability to cross-reference elements with (amongst other documents) a design specification. Loucopoulos, furthermore gives the criterion *executability* which intends to validate the specification against some facts in the modeled UoD. Halpin gives the criterion *expressibility* that should measure the extent in which a conceptual modeling language can capture the conceptually relevant details about an application domain. This criterion is based upon the 100 % principle in ISO [24]. Halpin introduces the *simplicity and orthogonality* criterion which measures the extent in which constraints from a conceptual modeling language can be added to a specification or left out whenever necessary. Finally, Halpin gives the criterion of *semantic stability*, which refers to the number of changes that have to be made in a conceptual model in the face of a change in the application domain.

Many of these criteria are qualitatively defined, however it is (in principle) possible to verify if a specific conceptual model complies to them

2.4 Criteria for Requirements Specifications

In the field of requirements engineering quality requirements focus on the way of modeling. In table 4 we summarize results of the IEEE Std. [23], Hevner and Mills [20], Wieringa [50] and Zowghi and Gervasi [52].

Hevner and Mills discuss a requirements elicitation approach in which they consider a hierarchy in requirements determination. The transaction hierarchy on a given level must be *consistent* with its higher level parent transaction. This is similar to *internal consistency* in the terminology of Boehm [3]. *Requirement completeness* is defined in the Hevner and Mills framework as the extent in which all system requirements are captured [20: p.232]. *Requirements closure* is the extent in which a RS is an instance of the meta-model for the RS. Hevner and Mills define clarity in two ways. On one hand *clarity* means that the requirements must be understandable for the user and the system users. On the other hand a requirements specification must present requirements in a form that is appropriate for system developers.

Table 4. Criteria for Requirements Specifications

Author	**IEEE-830 (1998)**	**Hevner and Mills (1995)**	**Wieringa (1996)**	**Zowghi and Gervasi (2003)**
criteria	Correct	Consistency	Communicability	Consistency
	Unambiguous	Closure	Truth	Completeness
	Complete	Completeness	Completeness	Correctness
	Consistent	Clarity	Feasibility	
	Ranked		Verifiability	
	Verifiable		Maintainability	
	Modifiable			
	Traceable			

The *communicability* criterion given by Wieringa [50] is divided into the sub-criteria *understandability* and *unambiguity*, which can be found respectively in the Khwaja and Urban list of criteria and the IEEE-830 set of criteria. The *truth* criterion according to Wieringa can be divided into the criteria *validity* and *implementation-independence* (also mentioned in the Loucopoulos [33] set of criteria)). Wieringa defines *completeness* as the extent in which a RS describes all requirements. The next criterion of Wieringa is *feasibility* consisting of the sub-criteria *consistency* and *cost-effectiveness*. Wieringa defines the latter sub-criterion as*: '... that the cost of implementing the product is justified by the benefit that accrues from implementing it.'* [50:p.78][2]. The definition of the *verifiability* criterion according to Wieringa is in line with the IEEE STD 830 criterion by the same name. Finally, Wieringa gives the criterion *maintainability* that is fully in line with the definition of Khwaja and Urban [25].

In the IEEE recommended practice for software requirements specifications [23, p.4] the criterion that a software requirements specification (SRS) should be *correct* is interpreted as the extent in which the customer needs are correctly reflected in the SRS. This criterion is the same as the *correct* criterion in Khwaja and Urban [25: p.587][3]. The criterion that states that a SRS must be *unambiguous* in order to achieve this the IEEE recommended practice for SRS's suggest to include a glossary in a SRS, in which the terms that could have multiple meanings are precisely defined. The third criterion in the IEEE standard is *complete(ness)* defined as the extent in which all significant requirements are acknowledged and treated. The *consistency* criterion in the IEEE standard refers to *internal consistency* in the same way as Boehm [3] defined this criterion: *'An SRS is internally consistent if and only if no subset of individual requirements described in it, conflict.'* [23: p.6]. The criterion *ranked* in the IEEE standard means that every requirement must have an identifier that indicates the importance or stability of such a requirement. The criterion *verifiable* is defined as *'A requirements specification is verifiable if, and only if, there exists some finite cost-effective process with which a person or machine can check that the software meets the requirement.'* [23]. This criterion is similar to the criterion *validatable* in Khwaja and Urban. The criterion *modifiable* refers to the extent in which changes to the requirements can be made consistently, completely and easily and this criterion therefore resembles the *simplicity and orthogonality* criterion given by Halpin [19]. The last criterion in the IEEE standard is traceability and overlaps with the criterion by the same name given by Loucopoulos [33] and the criterion by the same name in Khwaja and Urban *' An SRS is traceable if the origin of each of its requirements is clear and if it facilitates the referencing of each requirement in future development or enhancement documentation.'* [23: p.8].

Zowghi and Gervasi [52] investigated the relationship between *consistency*, *completeness* and *correctness* of a requirements specification. Their definition of *consistency* is the union of the definitions of *(internal) consistency* by Boehm [3] and the IEEE Std. 630 [23] and the definition of *unambiguous* in

[2] This criterion goes beyond the application scope of the REM criteria in this article.
[3] The IEEE Std. 830 was one of the sets of criteria that was compared in the Khwaja and Urban [25] article.

the IEEE standard. Zowghi and Gervasi specialize their *completeness* criterion into *internal completeness* and *external completeness*. *Internal completeness* in their definition overlaps with *closure* from the Hevner and Mills set of criteria. *External completeness 'ensures that all of the information required for problem definition is found within the specification.'* [52: p.996]. Zowghi and Gervasi conclude that *external completeness* is impossible to define and measure because the only complete specification of something would be the thing itself. They argue that *sufficient completeness* can be achieved at best. Zowghi and Gervasi [52:p.997] define the criterion *correctness* as *'Correctness of a requirements specification describes the correspondence of that specification with the real needs of the intended users…'* this definition overlaps with Essink and Romkema's criterion: *Specification must be an expression of the real domain requirements.*

Many of these criteria are qualitative and some formal definitions (*verifiability, external completeness)* are not operational.

2.5 Literature proposals for RE quality enhancement

In their classic book on MIS, Davis and Olson [11: p.480] propose four strategies for determining information requirements: asking, deriving from an existing information system, synthesizing from characteristics of the utilizing system and, discovering from experimenting with an evolving information system. Browne and Ramesh [5] gave some techniques that address the shortcomings that were given by Davis and Olson [11: p.474]. They derived three general categories for the techniques that addressed the cognitive problems: *pre-elicitation conditioning*, *direct prompting techniques* and *indirect prompting techniques*. Browne and Ramesh conclude that informal (external) representation techniques should facilitate the interaction between analysts and users and help overcome background differences among them.

Flynn [14: p.137-139] gives four requirements acquisition methods: observation, analysis of existing system, analysis of desired system documentation and interview and questionnaire. Larsen and Naumann [29] carried out an experiment in which they compared the analyst's ability to discover user requirements as a function of the knowledge representation they used: *abstract* or *concrete*. They indicate that an abstract representation (in this study a 'logical' DFD) is not as effective as a concrete representation (in this study a 'physical' DFD).

Lee and Kim [31] studied the relationship between formalization of the stages in the information systems development life cycle and the overall success of the (management) information system. They empirically demonstrated the relationship between the formalization of MIS development and MIS success.

Browne and Rogich [6: p.228] divide prompting techniques into *context-dependent techniques* and *context-independent techniques*. They propose that context-independent techniques are the most suited to be used by analysts in the elicitation of user knowledge in general, because analysts will often be assigned to

analyze the requirements of business processes for which their substantive knowledge is limited [6: p.231].

Bubenko and Wangler [7] propose a number of different techniques for the knowledge acquisition task: Analyzing example forms and structured documents produced by end users, reverse modeling of existing databases, accepting application descriptions in natural language

Wetherbe [49] gives as a proposed solution to the shortcoming in (executive) information requirements elicitation in the past, that the systems designers must be encouraged : *'to use a cross-functional, joint application design that involves input from all key decision makers in the business process .*'[49: p.64-65).

Flynn and Warhurst [15] empirically investigated the validation process within requirements elicitation and concluded that during validation, analysts perceive users as being unable to express their requirements adequately, and analysts have to employ informal realistic examples to explain the specifications to the users because the users do not feel comfortable with method notations.

Ambrosio et al. state that *' To improve the conceptualisation of the UoD during the schema design process, the use of linguistics is necessary. '* [1:p.112].

Kim and March mention two aspects of validation: *' comprehension and discrepancy checking. Users must comprehend or understand the meaning of the model. Then they must identify discrepancies between the model and their knowledge of reality.'* [26: p.103].

Sutton [43: p.116] discusses the aforementioned notion of 'perception' in the sense that 'meaning' implies 'meaning to someone' and that any meaning is constructed by an observer and therefore it can not exist objectively.

Recently, the research field of knowledge management has emerged in which next to well-structured information also the concept of 'tacit-knowledge' is the subject of analysis [10: p.50]. Polanyi classifies knowledge into *tacit* knowledge and *explicit* knowledge: *' Tacit knowledge is personal, context-specific, and therefore hard to formalize and communicate. 'Explicit' or 'codified' knowledge refers to knowledge that is transmittable in formal, systematic language.'* [27].

Summarizing: various quality definitions as well as various proposals for achieving a required level of quality are suggested in the literature. What is missing in these suggestions is a unifying framework that firstly links process quality (way of modeling) to product quality (way of working, way of controlling) of requirements elicitation and secondly shows overlapping and/or missing quality aspects of a RE method. In the next section we will synthesize the criteria found in the literature into a coherent and consistent set of criteria that can be used to evaluate requirements elicitation methods.

3. Synthesis of Criteria in the literature into a coherent and consistent set of criteria for requirements elicitation

For a synthesis of the variety of criteria found in the literature we distinguish two dimensions, being the applicability in the business UoD and the applicability in the Requirements Elicitation UoD.

Preferably a method for requirements elicitation has to be applicable in a wide range of (business) domains and it should be able to capture the whole range of requirements in any application area in which the REM is applicable. The extent to which a specific approach can be used in different types of (business) domains is an important quality criterion for a REM since it determines to a large extent the expected success of its application in terms of its outcome. We will refer to this criterion as *domain richness*. Furthermore application of a REM should lead to a *complete* RS for all application areas in which it is supposed to be *applicable.* We will refer to this criterion as *completeness.* In a simplified 2*2 model domain richness and completeness can be placed in the right boxes of figure 1 below.

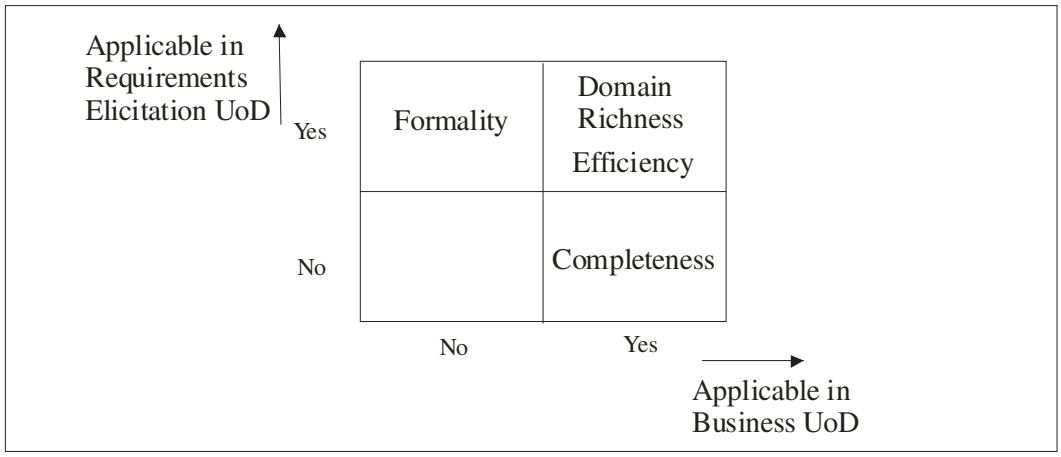

Fig. 1. Applicability of Quality Criteria for the RE and Business Application UoD

Once a RE-method is applicable, the remaining question is now *how* this complete RS can be derived in all possible application domains. From a managerial perspective it is paramount that RE processes should be executed in an efficient and controllable way. Efficient here means that the amount of (human and financial) resources that is needed to create such a RS must be minimal for any given quality level. In order to operationally use the result of an *efficient* RE-method, this result must be *formal*. We will refer to these two overarching criteria as *efficiency* and *formality.*

In this section we will show that we can map the partial and non-operational criteria that we have found in our literature survey onto the four mentioned criteria: *completeness, efficiency*, *formality* and *domain richness.*

Subsequently we will use this partition of the existing partial and non-operational criteria as starting point for a coherent and consistent set of operational criteria to which RE-method should comply. In figure 1 we have sketched the relevance of these criteria as a function of the RE product and the RE process. A set of criteria is coherent if they cover all relevant literature criteria, and it is consistent if its members (criteria) are non-overlapping.

3.1 The completeness criterion

In order to arrive at a requirements specification that contains all relevant domain semantics for the specification of a business application domain we first need to establish an idea of what we mean by *completeness* in the context of the way of modeling, the way of working and the way of controlling.

We will operationalize the completeness criterion for a requirements elicitation method, e.g. *what* must be incorporated in a requirements specification for an application domain. Olle et al. [39: p.41-43] distinguish three perspectives: the *data-oriented* perspective, the *process-oriented* perspective and the *behaviour-oriented* perspective. The data-oriented perspective should concentrate on the business data and must capture the domain concepts, the definition and the naming conventions for those domain concepts, the semantic relationships between the domain concepts and other 'static' and 'structural' knowledge in the enterprise. The process-oriented perspective should be able to capture the business activity and user perceivable tasks and describe what procedures exist for the creation of application facts or instances of semantic relationships. Finally, the behaviour-oriented perspective [39: p.43] should describe how 'events' can be cross-referenced to 'elements' in the process- and data-oriented perspectives [39: p.43]. This means that any requirements specification should potentially consist of models that cover these three (conceptual) perspectives.

Table 5. Types of rules versus perspectives [34, 39]

	state	**state**	**action**
Data-oriented	Data model	Static constraints	Dynamic constraints
Process-oriented		Static derivation	
Behavior-oriented			Dynamic rules

Loucopoulos and Layzell [34: p.264] consider two perspectives a *state* perspective and an *action* perspective and distinguish as core of their concepts a *data* model and a number of *rule types* . From the requirements specification point of view we can now link the Olle et al. three perspectives framework and the constraint typology by Loucopoulos and Layzell in table 5.

We can now conclude that the way of modeling for a requirements elicitation approach can be complete on two dimensions: the *number of perspectives* that are 'covered' by an approach and secondly, the *types of rules* within every perspective that can be encoded using this approach. With respect to the way of working, the *how* question is relevant, e.g. how do we find instances of these modeling constructs in a specific requirements elicitation project. The

availability of (modeling) procedures is of great importance here. In table 6 we have given the definition of the *completeness* criterion for the way of modeling and the way of working. For the way of modeling we have synthesized the definitions of *expressive adequacy* [25], *semantic relevance* and *abstraction mechanism* [19] and *ranked* [23]. For the way of working we have synthesized the criteria: *tolerance for incompleteness* and *multiple views* [25].

Table 6. The definition of the completeness criterion

	Way of modeling	**Way of working**
Definition of completeness criterion	The availability of conceptual modeling constructs for the data model, the static constraints, the static derivation, the dynamic constraints and dynamic rules that allow the modularization of the requirements specification .	The availability of procedures for instantiating the data model, the static constraints, the static derivation, the dynamic constraints and dynamic rules for the lowest to the highest level of specification completeness.

Definition 1. The completeness of a *requirements elicitation method* is the extent in which the completeness criteria in table 6 defined for the way of modeling, the way of working and way of controlling are satisfied.

3.2 The efficiency criterion

The operationalizing of the efficiency criterion for the purpose of evaluating requirements elicitation methods must take place for the way of modeling, the way of working and for the way of controlling. With respect to the requirements specification language (or way of modeling) we can say that the existence of more than one modeling construct that serves the same purpose has a negative impact on the efficiency of requirements elicitation process (or the way of working). Firstly, this relates to cognitive aspects of the resources needed to learn a specification language and secondly to the resources needed for selecting one modeling construct out of the set of alternative modeling constructs during the requirements elicitation process. Thirdly, the availability of 'equivalent' modeling constructs and the limitations under which they can be applied might lead to modeling rework in a later stage of the project when additional information about the requirements specification becomes available.

With respect to the efficiency in the way of working we can furthermore say that the availability of a (set of) procedure(s) that guide(s) an analyst in the requirements elicitation process will lead to an improved usage of (human) resources because it prescribes how an analyst should proceed in the process given the knowledge he/she has elicited so far [21] including stopping criteria [40]. Such a procedure, therefore, will minimize the required number of analysis steps and potential rework. In IEEE Std. 830 [37] and Khwaja and Urban [25], a criterion called *verifiable* is given in the context of software requirements specifications: *'A requirement is verifiable if, and only if, there exists some finite*

cost-effective process with which a person or machine can check that the software product meets the requirement.' [23]. An efficient procedure must contain a role definition [51: p.300] for the analyst and must clearly make a distinction between the responsibilities of the analyst and the responsibilities of the user.

With respect to the way of controlling we can define efficiency in two areas. Quality deficiencies in a RS must be prevented from happening, and if they do occur, they have to be 'repaired' by the process that is responsible for creating the deficiency. This means that the way of working of the REM must contain a number of 'quality-checking' verification sub-procedures, in such a way that the process that is responsible for the performance of a requirements elicitation activity is responsible for the assurance of its quality [18]. Further the way of controlling is concerned with the *project management* of the requirements elicitation project. The efficiency of project management must be measured in terms of the three project management targets: *performance, cost* and *time* [36: p.5]

Table 7. The definition of the efficiency criterion

	Way of Modeling	**Way of Working**	**Way of Controlling**
Definition of efficiency criterion	Average number of modeling constructs (of an average expression power) in a requirements specification language must be as low as possible for a given minimum required level of specification organization and for a given minimum required level of semantic stability. The modeling constructs should be easily learned and remembered.	Availability of procedure(s) that can be easily applied by an analyst and that will result in a maintainable specification.	Availability of quality assurance steps and the extent in which performance, cost and time can be optimized by having validation mechanisms for domain experts and the presence of go/no go controls in combination with communication milestones built into the modeling procedure(s)

In table 7 we have given the definition(s) for the efficiency criterion in which we have synthesized the following criteria for the way of modeling: *specification organizational support* [25], *semantic stability, simplicity and orthogonality* [19]. This definition balances the quest for a small set of very rich modeling constructs (which could be seen as ideal if requirements were stable) and the quest for a large set of easy to reassemble basic modeling constructs (which could be seen as ideal for maintenance). With respect to the way of working the following criteria have been synthesized into our definition: *Constructability, ease of use, help support*, and *support for maintainability* [25]. For the way of controlling we synthesized the criteria *communication* and *control* [51] and *validation mechanisms* [19].

Definition 2. The *efficiency* of a *requirements elicitation method* is the extent in which the efficiency criteria in table 7 defined for the way of modeling, the way of working and way of controlling are satisfied.

3.3 The formality criterion

A REM must lead to an *(internally) consistent* and *precise* requirements specification. In order to achieve requirements specifications that comply with these criteria we need a certain amount of formality in the way of modeling of the REM. *Formality* as a criterion can be found in 4 out of the eleven sets of criteria that we have found in the literature. Firstly, the modeling constructs that are used for the specification of requirements in the different perspectives must be formally defined. Secondly, the way of working, must be formalized in some sort of algorithm(s) that precisely prescribe(s) how the formal modeling constructs can be instantiated in order to obtain consistent specifications. These algorithms must contain facilities to question user assumptions regarding the domain knowledge.

Table 8. The definition of the formality criterion

	Way of Modeling	**Way of Working**	**Way of Controlling**
Definition of formality criterion	Extent in which modeling constructs in language are formally defined and can be used to create consistent and unambiguous specifications.	Extent in which procedure is formal in terms of its ability to provide internal verification support or closure and its ability to facilitate external validation.	Extent in which activities can be formally planned. Extent in which quality management is contained in formal (sub)procedure Extent in which provisions that enable traceability are contained in REM. Extent in which results of the RE process can be verified.

With respect to the way of controlling we must be able to formalize the planning of activities, for example in a precedence diagram. Boehm [3] defines *traceability* as a sub-criterion within consistency *('items in the specification have clear antecedents in earlier specifications or statement of system objectives.'* [3: p.78)). In the IEEE Std. 830 [23] *traceability* is defined as *backward* traceability (*'this depends upon each requirement explicitly referencing its source in earlier documents.'* [23: p.8]) and *forward* traceability (*'this depends upon each requirement in the SRS having a unique name or reference number.'* [23: p.8]). We conclude that quality assurance steps must be embedded in (a) formal (sub) algorithm(s) including provisions that enable traceability. We have synthesized the criteria *consistency* [20, 23, 51, 52] and *unambiguous* [23] into the definition of formality for the way of modeling. With respect to the way of working we have synthesized the criteria *executability* [33], *internal verification support, external validation support* [25], *closure* [20]. We have synthesized the *correctness* [23, 52] and *verifiable* [23] criteria into the formality definition for the way of controlling.

Definition 3. The *formality* of a *requirements elicitation method* is the extent in which the formality criteria in table 8 defined for the *way of modeling, the way of working* and *way of controlling* are satisfied

3.4 The domain richness criterion

The results of the literature review suggest a number of dimensions that determine the characteristics of the application domain and the scope of the requirements elicitation process. Before we go any further we have to define the paradigm that we will adhere to in the remainder of this article. We will adhere to the 'functionalism' paradigm [22: p.1202-1203] in which systems development and therefore requirements elicitation is considered to *'proceed from without, by application of formal concepts through planned intervention with rationalistic tools and methods.'* [22: p.1210]. Within this metaphysical position of functionalism we can characterize the domain requirements along four dimensions.[4]

The dimension *perception* refers to the extent in which different domain users have a different perception of an underlying reality. With respect to the dimension *turbulence* we can consider a situation in which the environment of the application domain is constant (nothing ever changes) and on the other side of the continuum an environment where there is continuous change.

Another dimension regarding the type domain under consideration is concerned with the extent in which the domain knowledge is 'tacit' versus 'explicit'. We will call this dimension the 'tacitness' dimension of the application

---

[4] The (requirements specification) technique criterion that was called *extent of applicability* in Khwaja and Urban [25: p.592] and that measures the general applicability of requirements elicitation methods we will call *domain richness*.

Land [28] distinguishes four categories of relationships between an information system (as the result of a systems development process) and its organizational environment: The *unchanging* environment, in which the information requirements of the system are not changing during its lifetime, the *turbulent* environment, in which the requirements over the expected lifetime of the system are always changing. The *uncertain* environment, in which the requirements of the system are unknown or uncertain and the *adaptive* environment, in which the output of the system has an influence on the environment.

Galal and Paul [16: p.93] challenge the 'fixed-point stance' towards requirements elicitation for a number of reasons. First requirements do change during the development of an information system. Secondly, they state that the statements in a requirements document are inherently predictive. In case of the 'wrong' predictions, the requirements need to be adapted. Thirdly, the requirements are context specific.

With respect to the changeability of requirements, Sutton [43: p.116] concludes that: *' It is becoming recognized that it is more appropriate to see requirements definition as a periodic or even continuous process that feeds other processes of delivery and review that may never end.'*

Galliers and Swan [17] introduce a two-dimensional framework for information systems development. The first dimension is the objective (formal) versus the subjective (informal) dimension. The second dimension is the unitary versus the pluralist (multiple stakeholders) dimension

subject area. The tacitness of a subject area can range from fully 'tacit' in which no single knowledge-creating process can be made explicit to a fully 'explicit' area in which every knowledge generating process can (potentially) be made explicit [27].

The fourth domain richness dimension is the way in which the requirements elicitation process is anchored. This dimension can range from a fully *abstract* anchor in which 'open questions' should be posed to a tangible anchor in which *tangible* example forms and structured documents are used. Examples of requirements techniques that can be considered to be positioned somewhere on the 'tangible' side of this dichotomy are the analysis of forms and structured documents by end users [7], others talk about 'realistic' examples [15] or 'verbalizing forms' [44]. If we look at the abstract side of this dimension we encounter techniques like direct-prompting techniques, directed questions and what-if analysis [5]. We can now summarize the four 'domain richness' dimensions that characterize application domains in table 9.

Table 9. Dimensions that characterize the application domain

Dimension	**Low extreme**		**High extreme**
Perception	uniform for all users	-	Different for all users
Turbulence	no change	-	continuous change
Tacitness	fully tacit	-	fully explicit
Anchoring	tangible	-	abstract

We will now give a definition of the domain richness criterion for requirements elicitation methods. This criterion will reflect the extent in which the four dimensions can be accommodated by a single requirements elicitation approach at the same time.

Definition 4. The *domain richness* of a *requirements elicitation method* is the extent in which this method can be applied under the full range of values for the given dimensions *perception*, *turbulence, tacitness* and *anchoring* in table 9.

3.5 Summary of the quality criteria for a REM

We have synthesized the criteria from the literature into our framework in which 4 sets of criteria are distinguished for the 3 'ways': the way of modeling, the way of working and the way of controlling for a RE-method and summarized them in table 10.

Table 10. Synthesis of RE criteria from literature onto the quality criteria derived in this study

Criterion	Way of Modeling	Way of Working	Way of controlling
Domain richness		Extent of applicability [25]. User participation [13]. Clarity [20]	
Completeness	Expressive adequacy [25], Completeness [3, 38, 50]. Implementation independence [33, 50]. Constructability [33]. Semantic relevance, abstraction mechanisms [19]. Ranked [23].	Tolerance for incompleteness , multiple views [25]. Completeness [20, 23, 52].	
Efficiency	Specification organizational support [25]. Simplicity and orthogonality, semantic stability [19]. Modifiable [23]. Understandability [25, 50].	Constructability, ease of use, help support, support for maintainability [13, 25, 50]. Efficieny [13, 51].	Communication, Control [51]. Validation mechanisms [19].
Formality	Consistency [3, 20, 23, 50, 51, 52]. Formality [38]. Unambiguous [23, 50].	Level of Formal foundation, internal verification support, external validation support [25, 50]. Closure [20]. Formal foundation [19]. Formality, Executability [33]	Traceability [33]. Correctness [23, 52]. Verifiable, traceable [23, 50].

We consider this set of criteria as coherent because all (in our view) relevant criteria in the literature are mapped in this table. The explicit division into way of modeling, way of working and way of controlling ensures consistency. Note that this set is theoretically incomplete, due to empty cells. This is left for further research.

4. The evolution of requirement elicitation methods

The quest for a quality requirement elicitation method is manifest from the very beginning of the MIS field [11]. Also the need to define the quality of an information system has been advocated [3]. Although it was tacitly assumed that some linkage between the two concepts of quality should exist (quality methods are the basis for quality systems) the overall pattern of the development of requirement elicitation methods shows fragmentation. There were typical schools to stress the importance of controlling the system development (e.g. SDM [45],

ISAC [35]), to stress the importance of the underlying data model (e.g. NIAM [47], (E)ER [9]), to stress the importance of user acceptance (e.g. ETHICS [37]) and to stress the strategic importance of an MIS (e.g. ISP [2]).

It is worthwhile noticing that these methods all either evaluated into methods with complex data models or became almost obsolete in the MIS field. The historical line from relational modelling to object oriented modelling shows an increased incorporation of business constraints in the modelling process. Objects, roles and constraints are supposed to capture business knowledge and this capturing is seen as mandatory for a quality requirement elicitation method. Today UML and NLM are methods with the most advanced modelling constructs.

From this observation it can be conjectured that traditional methods do not and modern methods do comply with the synthesized quality criteria as derived in this paper. For an overview and a detailed assessment of NLM to these criteria see [4].

5. Conclusions

Many traditional RE methods lack operational quality criteria. As a consequence the way of modeling and the way of working are intertwined and a formal starting point and formal ending point for a RE process is not defined.

Quality criteria from the literature (which are in principle applicable to RE methods) refer either to the way of modeling (the product of RE) or to the way of working (the process of RE) and do generally not or only partially address the way of controlling (project management).

The quality criteria from the literature can be synthesized into a coherent and consistent set of quality criteria for requirements elicitation methods. There are four criteria *completeness, efficiency, formality and domain richness* that apply to three aspects of RE, being the way of modeling, the way of working and the way of controlling.

The historical development of requirement elicitation methods (from EER to UML/NLM) shows a growing compliance to these criteria.

References

1. Ambrosio, A., Metais, E., Meurier, J-N. The linguistic level: Contribution for conceptual design, view integration, reuse and documentation. *Data & Knowledge Engineering 21*, (1997),111-129
2. Baker, B. The role of feedback in assessing information systems planning effectiveness. *Journal of Strategic Information Systems 4(1)* (1995), 61-80
3. Boehm,B. Verifying and validating software requirements and design specifications. *IEEE Software*, (January 1984), 75-88
4. Bollen, P. *On the applicability of requirements determination methods*. Ph.D thesis, faculty of Management and Organization, Rijksuniversiteit Groningen (2004)

(can be downloaded from http://www.ub.rug.nl/eldoc/dis/management/p.w.l.bollen/)

5. Browne, G., Ramesh, V. Improving information requirements determination: a cognitive perspective. *Information & Management, 39*, (2002),625-645

6. Browne, G., Rogich, M. An empirical investigation of user requirements elicitation: comparing the effectiveness of prompting techniques. *Journal of Management Information Systems, 17*, 4 (2001) ,223-249

7. Bubenko, A., Wangler, B. Research Direction in Conceptual Specification Development. In: Conceptual Modeling, Databases, and Case (Loucopoulos, P., Zicari, R. (eds.)). Wiley, (1992) 389-412

8. Byrd, T., Cossick, K., Zmud, R. A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Quarterly*, (march 1992), 117-138.

9. Chen, P. The entity-relationship model: Towards a unified view of data. ACM *Transactions on Database systems 1*, 1 (1976),9-36

10. Coughlan, J., Macredie, R.Effective Communication in Requirements Elicitation; A Comparison of Methodologies. *Requirements Engineering, 7*, (2002), 46-60

11. Davis, G., Olson, M. Management Information Systems, conceptual foundations, structure and development, 2nd edition, McGraw-Hill, (1985)

12. Dieste, O., Genero, M., Juristo, N., Mate, J., Moreno, A. A conceptual model completely independent of the implementation paradigm. *Journal of systems and software, 68*, (2003),183-198

13. Essink, L., Romkema, H. *Ontwerpen van informatiesystemen*. Academic-service., (1989) (in dutch)

14. Flynn, D. *Information systems requirements determination and analysis*. McGraw-Hill, (1992)

15. Flynn, D., Warhurst, R. An empirical study of the validation process within requirements determination. *Information Systems Journal, 4*, (1994), 185-212

16. Galal, G., Paul, R. A qualitative Scenario Approach to Managing Evolving Requirements. *Requirements Engineering 4*, (1999), 82-102.

17. Galliers, R., Swan, J. There's more to information systems development than structured approaches: information requirements analysis as a socially mediated process. *Requirements Engineering, 5*, 2 (2000), 74-82

18. Grunbacher, P., Halling, M., Biffl, S., Kitapci, H., Boehm, B. Integrating collaborative processes and quality assurance techniques: experiences from requirements negotiation, *Journal of Management Information Systems, 20*, 4 (2004), 9-29

19. Halpin, T. *Information modeling and relational databases: from conceptual analysis to logical design.* Morgan Kaufmann, (2001)

20. Hevner, A., Mills, H. Box-structured requirements determination methods. *Decision Support Systems, 13*, (1995), 223-239

21. Hickey, A., Davis, A. A unified model of requirements elicitation. *Journal of Management Information Systems*, *20*, 4, (2004),:65-84

22. Hirschheim, R., Klein, H. Paradigmatic Influence on IS development methodology: evolution and conceptual advances. In: *Advances in Computers*, Yovits, M. (ed.). JAI press London., (1992)

23. IEEE Std 830. *IEEE recommended practice for software requirements specifications.* Software engineering standards committee of the IEEE computer society, (1998)

24. ISO. Concepts *and terminology for the conceptual schema*, Griethuysen, J. et al. (eds.). ISO/TC97/SC5/WG3, (1982)

25. Khwaja, A., Urban, J.A synthesis of evaluatikon criteria for software specifications and specification techniques. *International Journal of Software Engineering and Knowledge Engineering, 12*, 5, (2002), 581-599

26. Kim, Y-G., March, S. Comparing Data Modeling Formalisms. *Communications of the ACM 38*, 6, (1995),103-115.

27. Kim, T-G., Yu, S-H., Lee, J-W. Knowledge strategy planning: methodology and case. *Expert Systems with Applications*, *24*, 3, (2003), 295-307

28. Land, F. A Contingency Based Approach to Requirements Elicitation and Systems Development. *Journal of Systems and Software, 40*, (1998),3-6

29. Larsen ,T, Naumann, J.An experimental comparison of abstract and concrete representations in systems analysis, *Information and Management, 22***,** 1, (1992), 29-40

30. Laudon, K., Laudon, J. *Essentials of management information systems*, 5th edition. Prentice-Hall, (2003)

31. Lee, J., Kim, S-H. The relationship between procedural formalization in MIS development and MIS success. *Information & Management*, *22*, (1992), 89-111

32. Leung, C., Nijssen, G. From a NIAM conceptual schema onto the optimal SQL relational database schema. *Australian Computer Journal***,** *19***,** 2 (1987), 69-75

33. Loucopoulos, P. Conceptual Modeling, in: *Conceptual modeling, databases and case: an integrated view of information systems development,* Loucopoulos, P, Zicari, R. (eds.), (1992)

34. Loucopoulos, P., Layzell, P. Improving information system development and evolution using a rule-based paradigm. *Software Engineering Journal, BCS/IEE, 4*, 5 (1989), 259-267.

35. Lundeberg, M. , Goldkuhl, G., Nilsson, G.: (A systematic approach to information systems development, *Information Systems 4,* (1979) 1-12 , 93-118.

36. Mantel, S., Meredith, J., Shafer, S., Sutton, M. *Project management in practice*. Wiley & Son, (2001).

37. Mumford, E. The ETHICS Approach. *Commun. ACM 36(6),* (1993), 82

38. Nijssen, G. On the gross architecture for the next generation database management systems. In: B. Gilchrist (ed.) *Information Processing 1977*, IFIP. (1977), p. 327-335.

39. Olle, T., Hagelstein, J., Macdonald, I., Rolland, C., Sol, H., Van Asche, F., Verrijn-Stuart, A.A. *Information Systems Methodologies- A Framework for Understanding*, North-Holland , (1988).

40. Pitts, M., Browne, G. Stopping behavior of systems analysts during information requirements elicitation. *Journal of Management Information Systems, 21*, 1 (2004), 203-226.

41. Rolland, C., Nurcan, S., Grosz, G. Enterprise knowledge development: the process view. *Information & Management, 36*, (1999), 165-184

42. Sinha, A., Popken, D. Completeness and consistency checking of systems requirements: An expert agent approach. *Expert systems with applications, 11*, 3 (1996), 263-276

43. Sutton, D. Linguistic Problems with Requirements and Knowledge Elicitation. *Requirements Engineering*, 5, (2000), 114-124

44. Ter Hofstede, A., Proper, H., Weide, T. van der. Exploiting fact verbalisation in conceptual information modelling. *Information Systems, 22*, (1997), 349-385

45. Turner, W., Langerhorst, R., Hice, G., Eilers, H. Uijttenbroek, A. SDM, *System Development Methodology*, Pandata, North-Holland, (1987)

46. Van Belle, J-P. A proposed framework for evaluating generic enterprise models. *South Africa Computer Journal,* (2000)

47. Verheijen,G., van Bekkum J. NIAM: An Information Analysis Method. In: Verrijn-Stuart,A., Olle T., Sol H., (eds.): *Proceedings of IFIP TC-8 CRIS-1 conference*, North- Holland Amsterdam, (1982), 537-590

48. Verhoef, T. *Effective information modelling support*. Doctoral Dissertation. Technical University Delft, the Netherlands, (1993).

49. Wetherbe, J. Executive Information Requirements : getting it right. *MIS Quarterly*, *15*, 1 (1991), 51-65

50. Wieringa, R. *Requirements Engineering: frameworks for understanding*. Wiley, (1996)

51. Wysocki, R., Young, J. *Information systems management principles in action*. John Wiley and Sons, (1990)

52. Zowghi, D., Gervasi, V. On the interplay between consistency, completeness, and correctness in requirements evolution, *Information and Software Technology, 45*, (2003), 993-1009.